

## CLOUD ROBOTICS: Navegación de un vehículo autónomo en un entorno con obstáculos

Manuel Costanzo<sup>1</sup>, Marcos Boggia<sup>1</sup>, Ismael Rodriguez<sup>1</sup>, Armando De Giusti<sup>1,2</sup>

<sup>1</sup> Instituto de Investigación en Informática LIDI, Facultad de Informática, UNLP-CIC

<sup>2</sup> Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)  
{mcostanzo, mboggia, ismael, degiusti}@lidi.info.unlp.edu.ar

**Resumen.** Este trabajo presenta el desarrollo de una plataforma WEB en el Cloud y el despliegue de un sistema Multi-Robot, compuesto por un chasis de auto Robot Rover 2WD, con sensores integrados, como así también, un Robot cámara aérea, que emula la captura de imágenes desde un cuadricóptero/drone, conectados al Cloud público de Amazon Web Services (AWS). Se detallan los prototipos desarrollados y el protocolo de comunicación utilizado; se mencionan los algoritmos implementados para el procesamiento y reconocimiento de imágenes, la simulación y determinación del camino óptimo, y la detección de obstáculos y espacios transitables, con el objetivo de que el vehículo no tripulado pueda llegar a un punto de destino de forma autónoma.

**Palabras Clave:** Auto Autónomo, Cloud Robotics, IoT, MQTT, Node-RED.

### 1 Introducción

Los avances en los paradigmas de Cloud Computing y Cloud Robotics han provocado un factor disruptivo de las TI en la industria tecnológica de sistemas Multi-Robot [1,2,3]. Estas tecnologías permiten tratar los datos de los componentes de hardware de un robot (sensores, actuadores/motores, cámaras, memoria, etc.), sin importar las limitaciones de cómputo y almacenamiento de las placas de desarrollo. En otras palabras, permite a los robots obtener resultados de tareas de cómputo intensivo, tales como: procesamiento de imágenes, determinación de rutas, confección de mapas, acciones cognitivas, etc., sin tratamiento local, sino en el Cloud [4].

Estos paradigmas brindan la capacidad de establecer escenarios para sistemas de Multi-Robot, donde cada robot se integra de un hardware mínimo con conectividad WiFi, donde los datos de los sensores y la adquisición de imágenes se procesan en el Cloud, y los actuadores de cada robot realizan las operaciones necesarias [5].

El propósito de investigación del presente trabajo es, el despliegue de un sistema de Multi-Robot, conectados al Cloud Público de AWS [6] con el fin de simular y ejecutar la navegación autónoma de un vehículo de 4 ruedas, como así también, la gestión de una cámara aérea de video, ambos conectados al Cloud, donde se realiza el procesamiento de imágenes y la planificación de caminos autónomos para que el vehículo efectúe la navegación a un punto de destino establecido.

Para ello, se determinó el método de comunicación con el Cloud; se registraron los robots en el servicio de AWS IoT [7]; se desarrollaron los algoritmos para

procesamiento y reconocimiento de imágenes, para simulación y determinación del camino óptimo al punto de destino establecido, y para detección de obstáculos y espacios transitables, con un bajo costo computacional.

A continuación, el presente trabajo se divide de la siguiente manera: en la sección 2, se introducen algunos conceptos elementales; en la sección 3, se presentan los prototipos confeccionados y sus componentes; en la sección 4, se describe el trabajo experimental realizado. Por último, la Sección 5, expone los resultados obtenidos y las líneas de trabajo futuras en relación a esta investigación.

## 2 Conceptos elementales

En esta Sección, se mencionan algunos conceptos empleados en la investigación realizada.

- **Amazon Web Services**

Es una plataforma de Cloud Público, que provee “Infraestructura como servicio”; a través de la tecnología de virtualización. De sus servicios más destacados, utilizamos “Elastic Compute Cloud” (EC2) [8] y “AWS Internet of Things” (AWS IoT).

- **Internet of Things (IoT)**

Es un nuevo paradigma que considera a los dispositivos como “objetos” conectados a Internet, utilizando protocolos de comunicación estándar [9].

- **Protocolo Message Queuing Telemetry Transport (MQTT)**

Es un protocolo de comunicación ligero [10,11], especialmente diseñado para tolerar conexiones intermitentes y reducir los requisitos de ancho de banda de red; soporta comunicación segura con TLS y calidad de servicio (QoS).

Implementa la comunicación de mensajes por medio de la publicación/suscripción sobre un canal de comunicación (tópico), como se puede observar en la figura siguiente:



Figura 1. MQTT: publicación/suscripción.

- **AWS IoT**

Es un servicio que proporciona AWS con el fin de conectar, administrar y operar grandes conjuntos de dispositivos [12]. Para cada dispositivo, se emite un certificado y un par de llaves privada-pública, que junto al certificado de la Entidad Certificante (CA), permite conectar el dispositivo al servicio de AWS IoT, en forma segura.

- **Node-RED**

Es una herramienta de programación visual, que permite programar algoritmos basados en flujos para IoT, sin la necesidad de escribir código. Provee de un editor de flujo basado en navegador WEB, que brinda una amplia paleta de nodos con diversa funcionalidad. El flujo se confecciona conectando los nodos entre sí. [13].

### 3 Prototipos y componentes

Para el presente trabajo se ha ensamblado un chasis de robot Rover de cuatro ruedas, emulando un vehículo tradicional de propulsión trasera, que cuenta con la capacidad de orientar las ruedas delanteras para la dirección del mismo. Se utiliza una placa micro-controladora de bajo costo y un módulo de expansión compatible, para controlar el motor, un sensor de ultrasonido y un módulo giroscopio para orientar al vehículo. Por otro lado, se realizó el despliegue de una cámara aérea que cuenta con una placa de desarrollo Raspberry Pi 3 y una cámara de video. A continuación se detallan los componentes:

Componentes	Descripción
<b>Chasis Rover 2WD con servo S3003</b>	Chasis compuesto de dos capas de acrílico, con un motor de 3 a 24v DC y velocidades con carga de 258 a 347 rpm/min, cuatro ruedas y un servo S3003 que permite direccionar el chasis con giros de hasta 45 grados. Dimensiones: 248mm (L) x 146mm (W) x 70mm (H).
<b>Microcontroladora Raspberry Pi 3</b>	Placa de bajo costo de desarrollo que brinda conexión WiFi, almacenamiento en memoria MicroSD de 32Gb y soporta alimentación de energía de 5v DC [14].
<b>RaspiRobot Board V3</b>	Módulo de expansión [15] compatible con Raspberry Pi 3, que brinda la interfaz de conexión entre la placa de desarrollo y los motores; incluye una fuente de alimentación conmutada para el suministro de energía, tanto de la placa Raspberry Pi como de los actuadores.
<b>Sensor Ultrasónico HC-SR04</b>	Sensor de distancias por ultrasonidos, capaz de detectar objetos y calcular la distancia a la que se encuentra, en un rango de 2 a 450 cm; cuantifica el delay de la señal, es decir, el tiempo que demora en regresar su eco.
<b>Fuente conmutada Step-Down</b>	Fuente de alimentación con regulador Step-Down DC-DC XL4015, que brinda una salida regulada de 1.25 a 36V.
<b>Acelerómetro y Giroscopio de 3 ejes</b>	Módulo Acelerómetro MPU, que posee un giroscopio de tres ejes con el que podemos medir velocidad angular, y un acelerómetro también de 3 ejes, con el que medimos los componentes X, Y y Z de la aceleración. Utilizamos el sensor de giroscopio para que el robot pueda posicionarse en la dirección correcta.
<b>Raspberry Pi Camera Board 2</b>	Pi Camera V2.1, es un módulo de video alta calidad de 8MP con sensor de imagen Sony IMX219 y un lente de foco fijo. Es capaz de tomar imágenes estáticas de 3280 x 2464 pixeles como así también, capturar vídeo de 1080p30, 720p60 y 640x480p90.

### 4 Trabajo experimental

El trabajo de investigación consistió en desarrollar una plataforma WEB que permite procesar, en el Cloud, los datos generados en un sistema de Multi-Robot que emula un vehículo de 4 ruedas y una cámara aérea de video, y planificar un camino autónomo para que el vehículo realice la navegación a un punto de destino.

El escenario está compuesto por el Robot Rover 2WD (RR2WD), obstáculos y un punto de destino; cada elemento es identificado por medio de una figura geométrica

(cuadrado, triángulo y círculo respectivamente). Quien se encarga de identificar cada componente es el Cloud, por medio de las imágenes provenientes del Robot cámara aérea. Una vez identificadas las figuras geométricas, se planifica el camino óptimo para que el vehículo se desplace hacia el punto destino sorteando los obstáculos.

A continuación, se describen los procedimientos para el procesamiento de imágenes y la determinación del camino óptimo.

#### 4.1 Procesamiento de imágenes

El procesamiento de imágenes se realiza en una instancia de EC2 en el Cloud, utilizando la librería OpenCV [16] y lenguaje de programación Python [17]. Se implementó un algoritmo que permite detectar e identificar sobre una imagen de captura aérea, figuras geométricas. Luego de la detección, se extraen las coordenadas de ubicación del RR2WD (cuadrado), el destino (círculo) y los obstáculos (triángulo) con el propósito de utilizar estos datos en una posterior planificación del camino óptimo.

#### 4.2 Cámara aérea

La cámara aérea emula la captura de video que puede realizar un drone. Dentro del escenario experimental se ubicó a una altura de tres metros sobre el nivel de la superficie del RR2WD, enfocando el escenario completo. Dicha cámara está integrada a una placa de desarrollo Raspberry Pi 3.

Cuando el Robot aéreo recibe una señal vía el protocolo MQTT, desde el Cloud, solicitando una captura de imagen, la misma se realiza en una resolución de 640x480 píxeles, se comprime y se codifica en base64, para por último enviarla vía MQTT a la instancia EC2, con el fin que el Cloud pueda realizar el reconocimiento y ubicación del RR2WD, los obstáculos y el punto de destino.

Todo el procesamiento mencionado se implementó con el lenguaje de programación visual Node-RED, como se muestra en la figura 2.

#### 4.3 Instancia EC2

Cuando la instancia EC2, recibe una imagen, vía MQTT, se procede a identificar las figuras geométricas en la misma. Como ya se ha mencionado, se debe detectar un “cuadrado blanco con bordes negros” para identificar al RR2WD, la figura geométrica de un “círculo blanco con bordes negros” para identificar al punto de destino; y un “triángulo blanco con bordes negros” que identifica los obstáculos.

Una vez detectadas las figuras, se procede a calcular el camino óptimo entre el vehículo y el destino. Previo a este procesamiento, es necesario haber calibrado la cámara.



**Figura 2.** Flujo de envío de imagen.

#### 4.4 Detección de figuras geométricas

Se implementó un algoritmo que detecta e identifica figuras geométricas dentro de una imagen. El proceso de detección consiste en transformar una captura habitual en una imagen en donde lo único que se pueda visualizar sean los vértices de las figuras en cuestión. Una vez transformada la captura, se identifican las figuras en función de la cantidad de bordes que tengan los objetos que sobresalen en la foto.

Cuando se recibe una captura aérea del escenario, se procede a transformar la imagen a escala de grises, como se muestra en la figura 3 y 4.



Figura 3. Imagen original.



Figura 4. Imagen en escala de grises.

Posteriormente, se suaviza la imagen como muestra la figura 5 y se aplica un umbral a la misma haciendo uso de la función de OpenCV “adaptiveThreshold” de la siguiente manera: “cv2.adaptiveThreshold (blurred, 255, 1, 1, 9, 5)”, para tener como resultado una captura como se visualiza en la figura 6.



Figura 5. Imagen suavizada.



Figura 6. Imagen post umbral.

Por último, en función de los contornos detectados, se aproximan los mismos y se detecta la figura geométrica con respecto a la cantidad de vértices que posea el contorno. Si la cantidad de bordes de un contorno es igual a 3, se identifica al mismo como un triángulo, si la cantidad de vértices es 4, se trata de un cuadrado/rectángulo, y si es mayor a 4, se identifica como un círculo.

##### 4.4.1 Calibración cámara aérea

Para lograr obtener una correspondencia entre el escenario real y el que muestra la imagen, se debe efectuar previamente una calibración de la cámara. Antes de la calibración se necesita conocer el ancho (en centímetros) de la figura geométrica del “cuadrado blanco con bordes negros” que representa al RR2WD.

El objetivo de la calibración es determinar la cantidad de centímetros contenidos en un pixel. Con esta referencia, se podrá conocer la distancia en centímetros entre cualquier par de píxeles, logrando así conocer la dimensión del escenario real.

La calibración se alcanza, calculando la distancia Euclidiana entre dos extremos de la figura geométrica del cuadrado, y se divide por el ancho de referencia de la figura anteriormente mencionada. Este procedimiento permite conocer cuántos pixeles forman un centímetro. Con esta información, se podrá calcular la distancia entre cualquier par de coordenadas (x,y) sobre la imagen.

#### 4.5 Planificación de camino óptimo

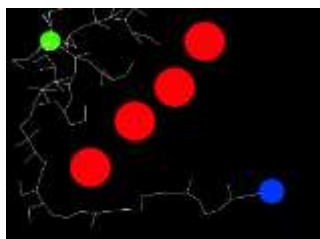
Una vez encontradas e identificadas las figuras geométricas en la imagen, se procede a planificar el camino óptimo. Un camino está compuesto por un conjunto de movimientos que guiarán al vehículo autónomo desde la posición donde se encuentra, hacia el punto de destino, evitando que colisione con los obstáculos que se presenten. El RR2WD puede realizar movimientos de giro en grados a derecha e izquierda (de 0 a 180 grados y de 0 a -180, respectivamente), como así también, movimientos para avanzar una determinada distancia en centímetros.

La planificación del camino consta de 3 fases: Búsqueda del camino, optimización del camino y transformación del camino.

##### 4.5.1 Búsqueda del camino

La técnica de planificación elegida es la denominada Informed RRT\* (Informed Rapidly Exploring Random Trees Star) (IRRT\*) [18]. Se implementó un algoritmo que se basa en la construcción de un árbol de configuraciones, que crece en forma aleatoria, con la búsqueda, a partir de un punto origen.

En la figura 7, se puede observar el resultado de simular la ejecución del algoritmo IRRT\* tomando como punto de inicio la ubicación del RR2WD (punto verde), el punto de destino (punto azul) y los obstáculos de por medio (puntos rojos). Como resultado de la planificación se obtiene un vector, donde cada posición contiene un nodo que representa una coordenada (x,y).

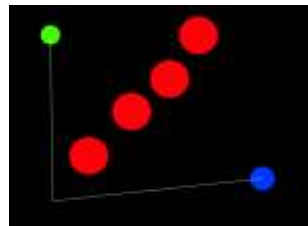


**Figura 7.** Simulación de la ejecución de la búsqueda del camino.

#### 4.5.2 Optimización del camino

Luego de encontrar las coordenadas del camino desde el inicio al destino, se procede a optimizar el mismo. Este algoritmo itera por los nodos resultantes del procesamiento de búsqueda IRRT\*, descartando aquellos puntos intermedios que no intersectan con un obstáculo, obteniendo como resultado un vector de coordenadas acotado.

En la figura 8 se puede observar el resultado luego de la optimización de la figura 7.



**Figura 8.** Resultado de la optimización del camino.

#### 4.5.3 Transformación del camino

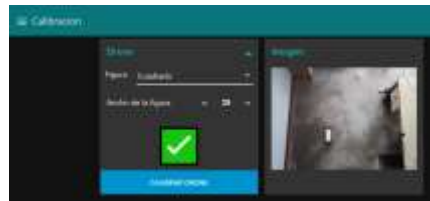
Para que el RR2WD pueda interpretar las acciones a realizar, se debe transformar el vector de coordenadas en movimientos. Cada movimiento consta de un grado de giro y una distancia en centímetros. Esta información se enviará al RR2WD para que realice la navegación autónoma hasta el punto de destino.

#### 4.6 Plataforma WEB

Se desarrolló una plataforma WEB basada en “Node-Red-Dashboard” [19] que brinda una interfaz gráfica, la cual permite realizar la calibración de la cámara, como así también, visualizar el resultado de la ejecución del algoritmo de planificación de caminos simulado. Por otro lado, permite dar inicio al sistema de Multi-Robot, mostrando en tiempo real todas las acciones que se estén llevando a cabo con el RR2WD, los resultados de la planificación de la trayectoria, y la imagen capturada de la cámara aérea.

#### 4.6.1 Calibración

Esta sección de la plataforma WEB, permite al usuario el ingreso de la información necesaria para calibrar las cámaras, por ejemplo: tipo de figura geométrica (“cuadrado” o “círculo”) y medida de ancho de la misma. Además, permite efectuar la calibración notificando con un mensaje de éxito o de error, según corresponda.



**Figura 9.** Interfaz de Calibración.

#### 4.6.2 Simulación

Esta sección brinda la posibilidad al usuario de simular la planificación de un camino, permitiendo ingresar un punto de inicio, un punto de destino y diversos puntos de obstáculos, dando a elegir el radio y color de los mismos, junto con opciones para eliminar los obstáculos y para resetear la simulación. Además, permite la posibilidad de visualizar el resultado final luego de la optimización o la animación de cómo el algoritmo de planificación RRT fue creando el árbol de nodos.



**Figura 10.** Simulación con animación.



**Figura 11.** Simulación sin animación.



#### 4.6.3 Interfaz en Tiempo Real (Live)

En esta última pantalla, se permite dar inicio y fin al sistema Multi-Robot, y visualizar en tiempo real la información que se está intercambiando y la situación en la que se encuentra cada actuador. Se puede apreciar gráficamente el resultado de la planificación del camino, y ver las imágenes que las cámaras están enviando a la instancia EC2 en el Cloud, como muestra en la siguiente figura:



**Figura 12.** Interfaz de gestión del sistema Multi-Robot.

## 5 Resultados obtenidos y líneas de trabajo a futuro

En esta investigación se ha implementado un sistema Multi-Robot conectado al Cloud público de AWS. Se han confeccionado los prototipos para emular un vehículo autónomo de 4 ruedas y una cámara aérea. Se han desarrollado los algoritmos que permiten realizar procesamiento de imágenes, reconocimiento de figuras geométricas y determinación del camino óptimo.

Además, se implementó una plataforma WEB para gestionar el sistema Multi-Robot, como así también, visualizar el resultado de simular la ejecución del algoritmo de planificación de camino óptimo IRRT\*.

Uno de los avances más significativos de esta investigación consiste en que, actualmente, el vehículo de cuatro ruedas conectado a AWS, se puede orientar, desplazar y arribar al destino establecido, de forma autónoma evitando los obstáculos del escenario.

Concluimos y reafirmamos que Cloud Robotics es una tecnología que favorece la evolución de los sistemas Multi-Robot, tendiente a reducir muchas de las limitaciones que se presentan en los sistemas de robots tradicionales.

En cuanto a las líneas de trabajo a futuro, se pretende integrar un drone en reemplazo de la cámara aérea fija, cuya navegación se ejecute en forma autónoma. Por otro lado, se espera adicionar uno o más robots Rover 2WD y 4WD, adaptando la solución para emular una ciudad en la cual los sistemas Multi-Robot puedan trabajar colaborativamente.

## Referencias

1. A. De Giusti, I. Rodríguez, M. Costanzo and M. Boggia, "Cloud Robotics: Auto Rover 4WD y Cuadricóptero controlados remotamente desde AWS" in CACIC '17. Proceedings of XVIII Workshop de Procesamiento Distribuido y Paralelo (WPDP) – XXIII Congreso Argentino de Ciencias de la Computación. La Plata, Argentina. 2017.
2. Costanzo M., Boggia M., Rodriguez I., De Giusti A.: "CLOUD ROBOTICS: Vehículo autónomo conectado a AWS". VI Jornadas de Cloud Computing & Big Data (JCC&BD 2018). La Plata, Argentina 2018.
3. L. Wang, M. Liu, M. Meng, R. Siegwart. "Towards Real-Time Multi-Sensor Information Retrieval in Cloud Robotic System". IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). Hamburgo, Alemania, 2012.
4. J. Kuffner. "Cloud-enabled robots". IEEE-RAS International Conference on Humanoid Robot. Nashville, USA, 2010.
5. L. Turnbull: "Cloud Robotics: Formation Control of a Multi Robot System Utilizing Cloud Infrastructure". Proceedings of IEEE – Southeastcon. Jacksonville, USA, 2013.
6. I. Bermudez, S. Traverso, M. Mellia and M. Munafò. "Exploring the cloud from passive measurements: The Amazon AWS case". Proceedings of IEEE – INFOCOM, 2013.
7. "AWS IoT" Available at: <https://aws.amazon.com/es/iot>. Accessed on 2018-06-06. [10] I. Bermudez, S. Traverso, M. Mellia and M. Munafò. "Exploring the cloud from passive measurements: The Amazon AWS case". Proceedings of IEEE – INFOCOM, 2013.
8. "Tipos de instancias de Amazon EC2" Available at: <https://aws.amazon.com/es/ec2/instance-types/>. Accessed on 2018-06-06.
9. L. Atzori, A. Iera and G. Morabito. "The Internet of Things: A survey". Journal Computer Networks ELSEVIER. Volume 54, Issue 15. 2010.
10. "Protocolo MQTT" Available at: <http://mqtt.org>. Accessed on 2018-06-06.
11. "OASIS MQTT", Available at: <https://www.oasis-open.org>. Accessed on 2018-06-06.
12. "AWS IoT" Available at: <https://aws.amazon.com/es/iot/>. Accessed on 2018-06-06.
13. "Node-RED" Available at: <https://nodered.org>. Accessed on 2018-06-06.
14. "The official site RapsberryPi" Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. Accessed on 2018-06-06.
15. "Raspirobot Board V3" Available at: <https://www.monkmakes.com/rrb3/>. Accessed on 2018-06-06.
16. "Open Source Computer Vision Library" Available at: <https://opencv.org/>. Accessed 2018-06-06
17. "Python Software Fundation" Available at: <https://www.python.org/>. Accessed on 2018-06-06.
18. Jonathan D. Gammell, Siddhartha S. Srinivasa and Timothy D. Barfoot. "Informed RRT\*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic" arXiv:1404.2334v3 [cs.RO] 28 Nov 2014.
19. "Node-Red-Dashboard" Available at: <https://github.com/node-red/node-red-dashboard>. Accessed on 2018-06-06.